Reinforcement Learning from Human Reward: Discounting in Episodic Tasks

W. Bradley Knox and Peter Stone

***Abstract*— Several studies have demonstrated that teaching agents by human-generated reward can be a powerful technique. However, the algorithmic space for learning from human reward has hitherto not been explored systematically. Using model-based reinforcement learning from human reward in goal-based, episodic tasks, we investigate how anticipated future rewards should be discounted to create behavior that performs well on the task that the human trainer intends to teach. We identify a "positive circuits" problem with low discounting (i.e., high discount factors) that arises from an observed bias among humans towards giving positive reward. Empirical analyses indicate that high discounting (i.e., low discount factors) of human reward is necessary in goal-based, episodic tasks and lend credence to the existence of the positive circuits problem.**

I. Introduction

Social rewards and punishments powerfully influence animal behavior, humans included. In recent years, this form of communication has been adapted to permit teaching of artificial agents by their human users [2], [16], [4], [15], [13], [10]. We call this form of teaching interactive shaping. Here, "human reward" is conceptually communicated to the trainer as signaling degrees of reward and punishment, approval and disapproval, or something similar, and the reward is received by the learning agent as a scalar value through varying interfaces (e.g., keyboard, mouse, or verbal feedback).

Interactive shaping enables people—without programming skills or complicated instruction—(1) to specify desired behavior and (2) to share task knowledge when correct behavior is already indirectly specified (e.g., by a pre-coded reward function). Further, in contrast to the complementary approach of learning from demonstration [1], learning from human reward employs a simple task-independent interface, exhibits learned behavior *during* teaching, and, we speculate, requires less task expertise and places less cognitive load on the trainer.

This paper is the first to assess a fundamental aspect of interactive shaping: how expectations of future human reward are discounted when an agent evaluates the quality of available actions. As we detail in Section II, past work on learning from human reward has consistently employed relatively high discount rates (some of which are previously unreported but were ascertained through email with the authors). This trend has gone unnoticed until now; this paper both identifies and justifies the trend. Besides being a curious aspect of past work, the question of discounting human reward is crucial because discounting directly determines what learning algorithms can be used and the flexibility of the agent (discussed in Section III). Additionally, the comparative analysis within this paper gives structure to the body of past work on learning from human reward, which previously lacked comparison between studies.

The authors are in the Department of Computer Science at the University of Texas at Austin. {bradknox,pstone}@cs.utexas.edu

We are generally interested in the question of how to perform reinforcement learning with human reward.[1] Reinforcement learning (RL) [14] without hidden state usually concerns solving tasks formulated as Markov Decision Processes (MDPs), denoted as $\{S, A, T, R, \gamma, D\}$. Here, S and A are the sets of possible states and actions; T is a function describing the probability of transitioning from one state to another given a specific action; R is a reward function, taking a state and an action as inputs; γ is a discount factor; and D is the distribution of start states for each learning episode. RL algorithms seek to learn policies ($\pi : S \rightarrow A$) for an MDP that maximize return from each state-action pair, $Q^\pi(s, a)$, where $Q^\pi(s, a) = \sum_{t=0}^{t} E_\pi[\gamma^t R(s_t, a_t)]$. We refer to such return-maximizing policies as *MDP-optimal*.

In this paper, we do not focus on designing algorithms that achieve MDP-optimal behavior. Rather, we investigate how to define an MDP such that MDP-optimal behavior performs well on the task the trainer intends to teach, as measured by a task performance metric τ. This problem is challenging because we, as algorithm designers, cannot specify the reward function, leaving that duty to the human trainer. But the discount factor can be controlled; we investigate the effect of this parameter in our experiments.

We specify MDPs in which the reward function is a predictive model of human reward, $\hat{R}_H : S \times A \rightarrow \mathbb{R}$, creating an MDP $\{S, A, T, \hat{R}_H, \gamma, D\}$. If an agent knows such an MDP, it may find the MDP-optimal policy, but that policy is *not* guaranteed to be the best possible policy according to τ. Indeed each choice of γ will lead to a *different* MDP-optimal policy. We ask the following experimental question: what discount factor maximizes the MDP-optimal policy's task performance, measured by τ?[2] In other words, we are exploring the space of the agent's objective, searching for the objective that when maximized leads to the best task performance (the trainer's objective).

This paper presents an application of model-based RL to learning from human reward (though this contribution is not our focus), where the reward function is learned from a human trainer and the transition function may be given, as it is in our experiments, and the agent plans with the two models. We find the model-based approach more informative than model-free RL because giving the agent knowledge of the MDP specification allows an agent to learn policies that perform well on the MDP more quickly,

making agent behavior more effectively reflect the current $\hat{R}_H$, approaching and often achieving MDP-optimal behavior that allows evaluation of the MDP specification itself.

We focus on episodic tasks [14] that are goal-based, meaning that the agent's task is to reach one or more goal states, after which the learning episode ends, a new episode starts with state chosen independently of the reached goal state, and the subsequently experienced reward is not attributable to behavior during the previous episode. As we explore throughout this paper, goal-based tasks have characteristics that make a comparison of different discounting rates especially informative. Despite our focus on goal-based tasks, however, we seek an algorithm that effectively learns in all episodic tasks, whether goal-based or not.

In Section II, past work is briefly reviewed. Section III discusses the consequences of the two extreme rates of discounting. Section IV presents a hypothesis about discounting—that *maximizing only immediate reward results in the best task performance*—and an intuitive argument for the hypothesis' likelihood that is built on observations that humans tend to give more positive reward than negative reward, creating what we term the positive circuits problem. In Section V, we describe two empirical analyses of discounting that support our hypothesis and the intuition behind it, after which we conclude the paper.

II. Past work on learning tasks from human reward

Interestingly, all previous algorithms have discounted more severely than is typical for MDPs. For episodic tasks, researchers have discounted by $\gamma = 0.75$ [16] and $\gamma = 0.9$ [15]. In continuing domains, $\gamma = 0.7$ [2], $\gamma = 0.75$ [13], $\gamma = 0.9$ [8], and $\gamma = 0.99$ [10] have been used.[3] The $\gamma = 0.99$ work is a non-obvious example of high discounting; with time steps of 5 ms, reward one second ahead is discounted by a factor of approximately 0.134. At the extreme of this trend, the TAMER framework discounts by $\gamma = 0$, learning a model of human reward that is (because of this discounting) also an action-value function [4]. This pattern of myopic maximization of human reward has hitherto not been identified.

In many of these studies, learning from human reward is shown to improve in some respect over learning only from MDP reward[4] (sometimes the championed learning algorithm uses both human and MDP reward and sometimes also a form of action suggestions) [16], [4], [5], [15]. In most of the others, learning from human reward is shown to be effective in a task where specifying an MDP reward function would be infeasible in the motivating use case [2], [10] (i.e., training a user-specific policy when the user cannot program).

III. Consequences of discounting

The two extremes of discounting have different advantages, briefly described in this section.

For $\gamma = 1$, the agent acts to maximize the undiscounted sum of future reward. With this discounting, the reward function could encode a trainer's desired policy, the trainer's idea of the task goal, or some mixture of the two; expression of a task goal permits simpler reward functions (e.g., 0 for transitions that reach the goal and -1 otherwise), which could reduce the need for training, allow the agent to find behaviors that are more effective than those known by the trainer, and make the agent's learned behavior robust to environment changes that render ineffective a previously effective policy but leave the purpose of the task unchanged (e.g., when the MDP-optimal path to a goal becomes blocked, but the goal remains unchanged). Given a model of system dynamics (i.e., a transition model) and a planning algorithm, these advantages become even more pronounced.

For $\gamma = 0$, the agent acts myopically to maximize immediate reward. This objective is simpler algorithmically, since a discount factor of zero reduces reinforcement learning to supervised learning. Supervised learning is generally an easier problem, and such discounting enables the agent to build upon a larger body of past research than exists for reinforcement learning, including tools for automatic selection of features, the representation of the human reward model, and the algorithm for learning parameters of this model. A disadvantage of this discounting, on the other hand, is that the reward model can encode a policy but not more general goals of the task.

Our ambition in this work is to create a natural interface for which people generate reward on their own. Accordingly, we observe that *algorithm designers should choose a discounting level that is compatible with human reward rather than assuming the human trainers will fit their reward to whatever discounting is chosen*. Granted, there appears to be some flexibility in the choice of algorithm: trainers can be instructed before they teach, and humans appear to adapt to the interface and learning algorithm with which they interact. But it may nonetheless be the case that certain intuitively appealing algorithms are incompatible with some or all human training, even after instruction and practice. The rest of this paper explores such a possibility.

IV. Intuition for incompatibility of human reward with $\gamma = 1$

In this section, we describe our intuition in two parts for why treating human reward identically to conventional MDP reward in episodic, goal-based tasks—i.e., using $\gamma = 1$—will often cause minimal task performance, a situation we call the positive circuits problem.

A. Humans tend to give more positive than negative reward

Thomaz and Breazeal conducted experiments in which humans train agents in an episodic, goal-based task [16]. Focusing on the first quarter of the training session, when the agent's task performance is generally worst, they found that 16 out of 18 of their subjects gave more instances of positive reward than of negative reward.

We also examined the balance of positive and negative reward from previous experiments, specifically from 27 subjects teaching TAMER agents to play Tetris (the control

condition of the "critique experiment" in [3]) and 19 subjects teaching TAMER agents to perform the mountain car task [4] (as defined in Sutton and Barto [14]). Comparing the sums of each trainer's positive reward values and negative reward values, they found that 45 of the 46 trainers gave more positive reward than negative over their training session.[5]

Based on past experiments, human trainers appear to generally give more positive reward than negative reward with remarkable consistency.

B. Consequences of positive reward bias for learning with large discount factors

In many goal-based tasks, there exist behavioral circuits that the agent can repeatedly execute, returning to the same or similar states. Such circuits exist for many MDPs, including any deterministically transitioning MDP with at least one recurrent state and any MDP that contains at least one state in which an agent can remain by taking some action. A simple example is an agent walking in circles in a navigational task. For such tasks, given the predominance of positive reward, it is likely that at least one such circuit will provoke positive net reward over each iteration of the circuit. Assuming that the goal-based task is episodic (i.e., a goal state is an absorbing state that ends a learning episode, a large class of problems), the MDP's discount factor γ is conventionally 1. Given that $\gamma = 1$, the expectation of return from states along a net-positive reward circuit will consequently be infinity, since the return is the sum of infinitely repeated positive reward. *Therefore, if a circuit exists with net-positive reward, an MDP-optimal policy for $\gamma = 1$ will never reach the goal*, since reaching absorbing state will end accrual of reward, making the return of a goal-reaching state-action pair finite, regardless of how large the reward is for reaching the goal. Thus, we call this issue the positive circuits problem. The general problem of positive circuits in RL has been discussed previously [11], [9] but to our knowledge has not been connected to human-generated reward or episodicity.

Positive circuits can also be problematic at high γ values that are less than 1. For instance, if $\gamma = 0.99$ and some circuit exists that has an average reward of 0.5 per transition, expected return from at least one state in this circuit will be approximately 50 or higher (because $\sum_{t=0}^{\infty} 0.5 * \gamma^t = 100$). Though finite, such high expectations of return may, despite the trainer's best efforts, be larger than the expectation of return for any path from the state to the goal.

Trainer adaptation may be insufficient to avoid such a goal-averse result; delivering reward such that there are zero repeatable circuits of positive net reward may be severely unnatural for a trainer. Consequently, we hypothesize that RL algorithms using $\gamma = 0$ for human rewards will generally perform better on the trainer's task performance metric τ on goal-based, episodic tasks.

V. Empirical analysis

In this section, we present two empirical analyses of the impact of different discount factors when learning goal-based, episodic tasks from human reward. Recall that, as discussed in Section II, maximizing the discounted sum of human reward is not equivalent to maximizing task performance. In fact, it is precisely the relationship between these two types of objectives that we are investigating.

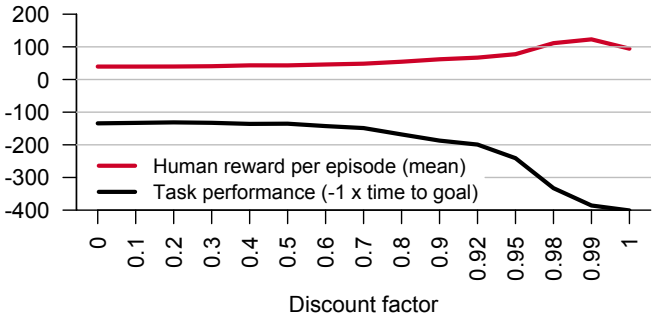

Fig. 1. Aggregate results for the γ-independent model analysis, showing mean task performance and mean sum of $\hat{R}_H$-generated reward per episode for mountain car, over the final 500 episodes of 4000 episodes of learning.

In both analyses, the model of human reward, $\hat{R}_H$, is learned through the TAMER framework [4], and the output of this model provides reward for the agent within an MDP specified as $\{S, A, T, \hat{R}_H, \gamma, D\}$. During training, $\hat{R}_H$ is updated by human reward signals. The agent seeks to maximize the expectation of the sum of $\hat{R}_H$'s future output from any given state, $Q^{\pi}(s,a) = \sum_{t=0}^{\infty} E[\gamma^t \hat{R}_H(s_t, \pi(s_t))]$, but the agent is evaluated by a task performance metric τ. From a start state, this return of predicted human reward is denoted $V_{\pi}(s_o)$. For both tasks used below, the conventional MDP specifications (i.e., with hard-coded reward functions) have $\gamma = 1$; thus, at $\gamma = 1$ $\hat{R}_H$ is being used as if it were interchangeable with a conventional MDP reward function.

During training for both analyses, human reward was given via two keys on the keyboard, which mapped to 1 and -1. This mapping, though not infallible, is an intuitive choice that is similar to that of related works that explain their exact mappings [16], [15], [10], [13].[6]

A. Varying γ with pre-trained human reward models

This first analysis uses 19 fixed $\hat{R}_H$s learned from the training logs created from a past experiment using TAMER [4], taken from the third run of 19 trainers of the mountain car task. In mountain car, a simulated car must accelerate back and forth across two hills to reach the top of one. Each of these $\hat{R}_H$s provide reward for an RL algorithm at various discount factors. We call this experiment the "γ-independent model experiment" because the human reward data was gathered under $\gamma = 0$ discounting, which differs from the discounting of most of our experimental conditions. We discuss possible training bias caused by such mismatched training and testing at the end of this section.

The RL algorithm is an enhanced Sarsa(λ) algorithm that exhaustively searches a transition tree up to 3 steps ahead.[7] For these experiments, the agents learn from $\hat{R}_H$ for 4000 episodes, and episodes are terminated (with an update of 0 reward) if the goal is not attained after 400 time steps,

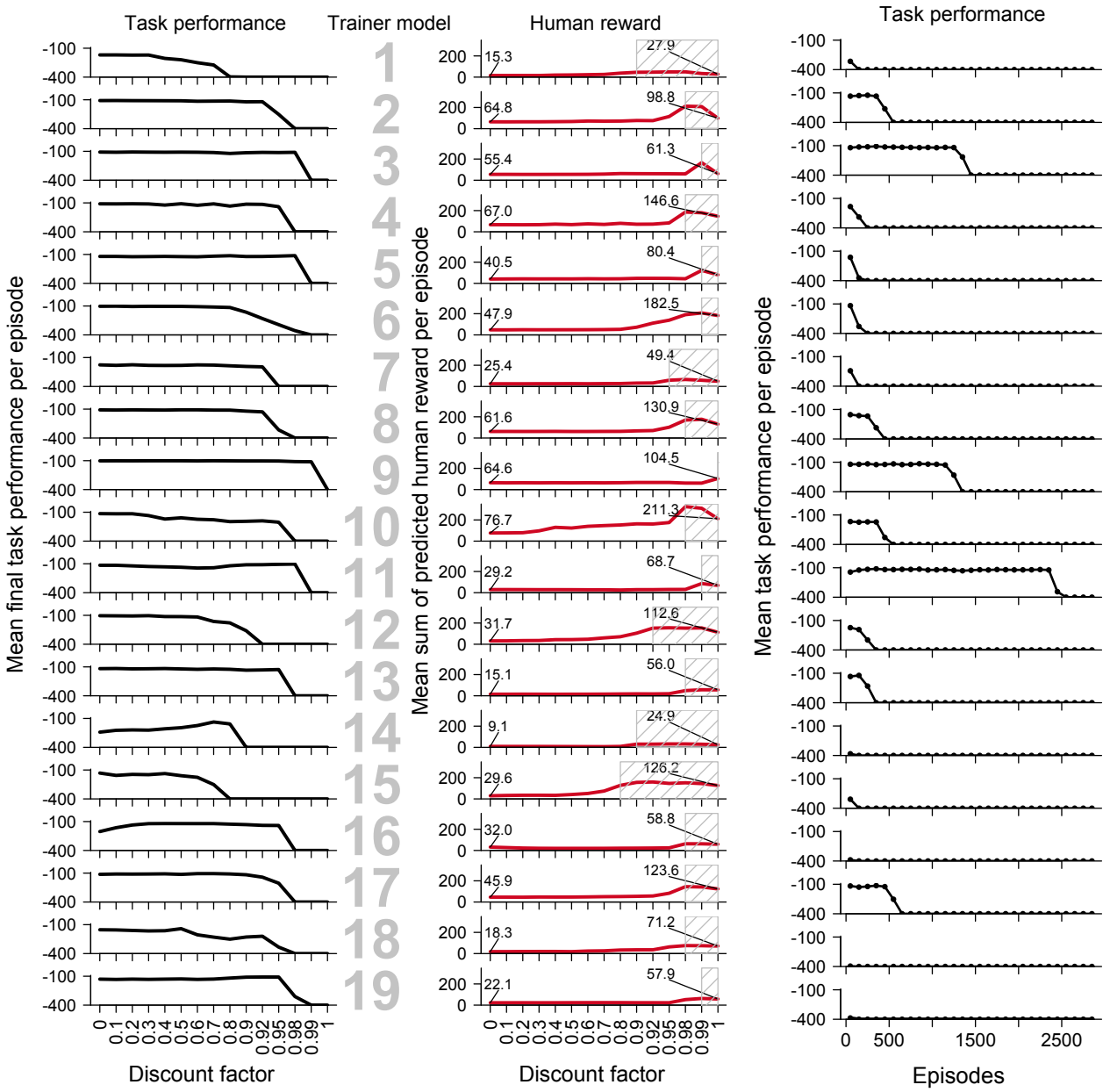

Fig. 2. (a) Non-aggregate version of the results in Figure 1, showing learned task performance over 500 episodes after 3500 episodes of learning (left) and mean total reward per episode over these final 500 episodes (right) for each of the 19 trainer models. Gray shading in the "Human reward" column indicates the inclusive range of γ values at which the agent's task performance is minimal (-400 per episode). (b) Learning curves at $\gamma = 1$, showing mean task performance over 100 episode intervals for each trainer model.

limiting the agent's maximum return to a finite value.

The $\hat{R}_H$s—the trainer models—were learned with the same linear representation over Gaussian RBF features that was used during the live training session, updating by incremental gradient descent [4].[8]

Figure 1 displays mean task performance and mean total reward per episode for each tested discount factor across all 19 $\hat{R}_H$ models. Additionally, Figure 2 displays the same data for each model separately to allow further inspection and to show the consistency of qualitative results between various models. We consider final performance to be over the last 500 episodes of learning.

Most importantly, at final performance all trainer models led to the worst possible return at $\gamma = 1$. With $\gamma = 0.99$, 18 models led to minimal return. We visually examined agents learning at $\gamma = 1$ from five of the trainer models; each agent exhibited a circuitous behavior, indicating the positive circuits problem is likely responsible for minimal task performance. Indeed, the mean sum of predicted human reward per episode increases as performance decreases, as can be seen in the plots of the final task performance with each trainer model (Figure 2). For all 19 trainer models, the

mean reward accrued per episode is higher at discount factor of 1 than 0. Further, for almost every trainer, at every γ value that leads to worst-possible task performance (i.e., values shaded gray in the "Human reward" column of Figure 2), the corresponding mean total reward per episode is higher than at all γ values that lead to better performance. The three exceptions (trainer models 2, 3, and 6) break this general observation by small margins, 15% or less.

Two general patterns emerge. We have noted the first: performance decreases as the discount factor increases. Secondly, agent algorithms also accrue higher amounts of predicted human reward as the discount factor increases. In other words, best task performance is not aligned with behavior that accrues the most predicted human reward.[9]

Figure 2(b) shows learning curves at 100-episode intervals for a single run at $\gamma = 1$ for each trainer model. Good initial performance lasts for a varying amount of time but then degrades to worst-possible performance quickly. In the plots, this degradation occurs during the intervals with intermediate performance.

There is one important caveat to the conclusions we draw from this γ-independent model analysis. Training occurred with TAMER algorithms, effectively at $\gamma = 0$. We strongly suspect that trainers adjust to the algorithm with which they interact; if the agent is maximizing immediate reward, a trainer will likely give more reward for immediately previous behavior. Only a γ-dependent model analysis—as we perform in the following experiment—will address whether this caveat of trainer adjustment has affected our conclusions.

B. Setting γ before training human reward models

In the analysis described in this section, as in the γ-independent model analysis in the previous section, the human reward model $\hat{R}_H$ is learned by TAMER and provides predictions that are interpreted as reward by an RL algorithm. But unlike the previous analysis, $\hat{R}_H$ is learned while performing reinforcement learning, and the RL algorithm selects actions while learning $\hat{R}_H$—not after $\hat{R}_H$ is learnt under TAMER's $\gamma = 0$ discounting. Thus the human trainer will be adapting to the same algorithm, with the same γ, that is being tested.

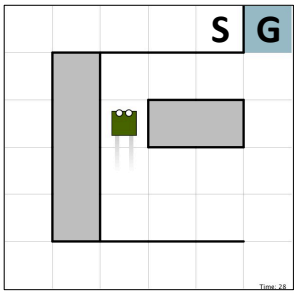

Fig. 3. A screenshot of the grid-world task used in the second experiment. To communicate the agent's actions and state transitions to the trainer, the simulated robot's eyes point in the direction of the last action and wheel tracks connect the agent's last occupied cell to its current location. Start and goal cells are labeled 'S' and 'G' respectively.

Because the agent in this experiment learns from a frequently changing reward function, behaving optimally with respect to the current reward function is difficult. Our choice of task and RL algorithm creates approximately MDP-optimal behavior with small lag in responding to changes to the reward function, a lag of a few time steps or less.

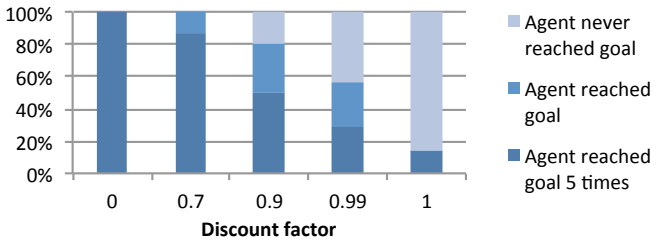

Fig. 4. Success rates for the grid-world experiment by discount factor.

The task is a grid world with 30 states, shown in Figure 3. At each step, the agent acts by moving up, down, left, or right, and it cannot pass through walls. Task performance is measured as the time to reach the goal. The agent always starts a learning episode in the state labeled "**S**" in Figure 3. The shortest path from the start state requires 19 actions. Each time step lasts approximately 800 ms.

The reinforcement learning algorithm is value iteration [14], except that instead of iterating until state values converge, one update sweep over all of the states occurs each 20 ms, creating 40 sweeps per step. At each step, the agent greedily chooses the action that maximizes predicted return for the current state, as calculated by a one-step lookahead for each action, using the predicted human reward and discounted value of the next state.

The TAMER module learns and represents the human reward model $\hat{R}_H$ as a linear model of Gaussian RBFs. One RBF is centered on each cell of the grid world, effectively creating a pseudo-tabular representation that generalizes slightly between nearby cells.[10]

The experiments were conducted through subjects' web browsers via Amazon Mechanical Turk. Subjects were randomly given an algorithm using one of five different discount factors: 0, 0.7, 0.9, 0.99, and 1. For these five conditions, the respective number of subjects was 10, 8, 10, 7, and 7.[11] Subjects were prepared with video instructions and a period of controlling the agent followed by a practice training session. The real training session stopped after the agent reached the goal 5 times or after 300 steps, whichever came first.

Figure 4 shows the success rate of trained agents by condition, dividing them among those that never reach the goal, reach the goal 1–4 times, and reach the goal the maximum 5 times. Task performance consistently worsens as the discount factor increases, a pattern supported by significance testing. Fisher's Tests compared outcomes of reaching the goal all 5 times or not by condition: between $\gamma = 0$ and $\gamma = 1$, $p = 0.0006$ (extremely significant); between $\gamma = 0$ and $\gamma = 0.9$, $p = 0.0325$ (significant); and between $\gamma = 0$ and $\gamma = 0.7$, $p = 0.4444$ (not significant).

To evaluate the intuition given in Section IV for why $\gamma = 1$ discounting might be problematic in a goal-based episodic task, we examine the ratio of cumulative positive reward to cumulative negative reward given by successful trainers in each condition, shown in Figure 5. Success appears highly related to this ratio; in Figure 5, we are

able to draw a dividing line at each condition between all agents that never reach the goal and all other, more successful agents. Additionally, the ratio of this division between success and failure monotonically decreases as the discount factor increases, which supports our conjecture that the positivity of human reward becomes more problematic as the discount factor increases (Section IV). Without recognition of the positive circuits problem (Section IV), this pattern of lower-performing agents getting more reward would be quite counter-intuitive. Further, negative Spearman correlations between discount factor and these ratios are extremely significant both for all trainers and for only trainers whose agents reached the goal once or more ($p <= 0.0005$), but the correlation when considering only goal-reaching trainers is stronger ($\rho = -0.7594$, compared to $\rho = -0.543$ for all trainers). We conjecture that γ affects ratios by both filtering out trainers that give too much positive reward in conditions of higher γs and by pressuring trainers to adjust their ratio in response to the agent. In surveys given after training, at least one trainer, from the $\gamma = 0.9$ group, spoke of his attempts to adapt to the agent: "When [the reward] key is stroked there is not much response in the robot. Only [the punishment] key stroke worked."

Reward is predominately positive (a ratio greater than 1) for 66.7% of trainers in this experiment. Though this result supports the conjecture that human reward generally has a positive bias, we do see a higher incidence of negative training than did past work (see Section II), mostly from higher γ values than had previously been reported.

After training, there was at least one behavioral circuit with net-positive reward in 35 of the 42 MDPs created from trainers' reward models. In other words, 83.3% of the trained agents would exhibit the positive circuits problem if learning with $\gamma = 1$. Half of the predominately negative trainers created positive circuits. Those without positive circuits all had positive-to-negative reward ratios below 0.63 and generally were from higher γ experimental groups: one from 0.7 and two each from 0.9, 0.99, and 1.

VI. Conclusion

Given a reward function—a model of human reward in this paper—the choice of γ determines the MDP-optimal policies. We investigate which γs create MDP-optimal policies that perform best on the task performance metric τ that the trainer seeks to maximize. The empirical results described in Section V indicate that MDP-optimal policies defined by low γ values (e.g., 0) translate to the best task performance in goal-based, episodic tasks. Consequently, *human reward cannot naively be learned from as if it is conventional MDP reward*, an approach that would entail high discount factors and is shown here to potentially lead to minimal task performance. Further, the results lend credence to the prevalence of the positive circuits problem (Section IV), our speculative explanation for the relationship between discounting and task performance: the positivity of human reward will lead to infinite, goal-avoidant behavioral circuits. More specifically, Section V-A demonstrates that raising the discount factor—with static human reward models that are successful at lower discount rates—can cause an RL agent to accrue more cumulative human reward while performing worse on the task. And in Section V-B, the rate of successful training and the ratio of the total positive reward to total negative reward by successful trainers monotonically decline as the discount factor increases.

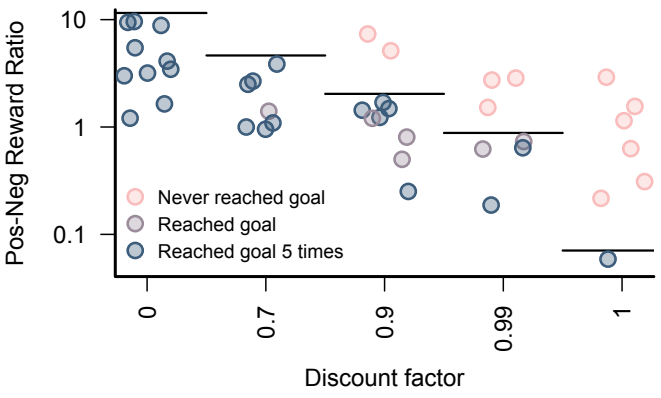

Fig. 5. Ratio of cumulative positive reward to cumulative negative reward given by each trainer, divided by discount factor condition and task performance. Jitter has been added along the x-axis for readability. For each condition, a horizontal line was placed above the mark for the highest ratio at which a subject trained the agent to reach the goal at least once.

Since TAMER, with $\gamma = 0$, has been implemented successfully to train episodic tasks that are *not* goal-based (in Tetris [4], keepaway soccer [12], and cart pole [7]), we suspect that further investigation will reveal that $\gamma = 0$ generally results in the best task performance in common episodic tasks, goal-based and otherwise.

This paper represents a step forward in the effort to create effective algorithms for learning from human reward. We note, however, that more analysis is required before one can decisively conclude that $\gamma = 0$ is ideal for learning from human reward, at which point the phrase "human reward" may need to be exchanged for terminology that does not confuse this form of feedback with reward from reinforcement learning. Changing the mapping of keys to scalar values, the instructions to trainers, and our algorithmic choices—though all carefully chosen to avoid overt bias—might create qualitatively different results.

Additionally, our argument that high discount factors can lead to infinite circuits—and thus minimal task performance—is specific to episodic tasks. An intriguing direction of inquiry, which we are currently undertaking, is whether the results will change if the task is made continuing in the eyes of the agent, possibly by creating an experienced transition between episode-ending states and start states.

Acknowledgments

This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (IIS-0917122), ONR (N00014-09-1-0658), and the FHWA (DTFH61-07-H-00030). We also thank

George Konidaris and Rich Sutton for fruitful discussions on discounting human reward.

REFERENCES

[1] Argall, B., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robotics and Autonomous Systems **57**(5), 469–483 (2009)

[2] Isbell, C., Kearns, M., Singh, S., Shelton, C., Stone, P., Kormann, D.: Cobot in LambdaMOO: An Adaptive Social Statistics Agent. AAMAS (2006)

[3] Knox, W., Glass, B., Love, B., Maddox, W., Stone, P.: How humans teach agents: A new experimental perspective. International Journal of Social Robotics, Special Issue on Robot Learning from Demonstration (2012)

[4] Knox, W., Stone, P.: Interactively shaping agents via human reinforcement: The TAMER framework. The 5th International Conference on Knowledge Capture (2009)

[5] Knox, W., Stone, P.: Combining manual feedback with subsequent MDP reward signals for reinforcement learning. Proceedings of The 9th Annual International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2010)

[6] Knox, W.B.: Learning from human-generated reward. Ph.D. thesis, Department of Computer Science, The University of Texas at Austin (2012)

[7] Knox, W.B., Stone, P.: Reinforcement learning with human and MDP reward. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2012)

[8] León, A., Morales, E., Altamirano, L., Ruiz, J.: Teaching a robot to perform task through imitation and on-line feedback. Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications pp. 549–556 (2011)

[9] Ng, A., Harada, D., Russell, S.: Policy invariance under reward transformations: Theory and application to reward shaping. ICML (1999)

[10] Pilarski, P., Dawson, M., Degris, T., Fahimi, F., Carey, J., Sutton, R.: Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning. In: Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on, pp. 1–7. IEEE (2011)

[11] Randløv, J., Alstrøm, P.: Learning to drive a bicycle using reinforcement learning and shaping. In: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 463–471. Citeseer (1998)

[12] Sridharan, M.: Augmented reinforcement learning for interaction with non-expert humans in agent domains. In: Proceedings of IEEE International Conference on Machine Learning Applications (2011)

[13] Suay, H., Chernova, S.: Effect of human guidance and state space size on interactive reinforcement learning. In: RO-MAN, 2011 IEEE, pp. 1–6. IEEE (2011)

[14] Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press (1998)

[15] Tenorio-Gonzalez, A., Morales, E., Villaseñor-Pineda, L.: Dynamic reward shaping: training a robot by voice. Advances in Artificial Intelligence–IBERAMIA 2010 pp. 483–492 (2010)

[16] Thomaz, A., Breazeal, C.: Teachable robots: Understanding human teaching behavior to build more effective robot learners. Artificial Intelligence **172**(6-7), 716–737 (2008)

NOTES

[1]We write "human reward" because we seek algorithms that capture the simplicity and ease of teaching by reward and punishment among natural organisms. Further, this usage is consistent with most past work, fits our inclusion of the words "reward" and "punishment" in instructions to trainers, and may be fully correct if human reward is ultimately found to be equivalent to some form of reward in reinforcement learning. We recognize that we may ultimately find that this form of feedback does not map to "reward" as it is used in reinforcement learning, necessitating a change of terminology.

[2]Though τ may be subjective and flexibly defined in practice, in this paper the trainer is given a static, pre-specified τ to maximize, facilitating empirical evaluation of the MDP's resultant task performance.

[3]The discount factors for three publications were learned through personal correspondence with an authors Isbell [2] and Morales [15], [8].

[4]As shorthand, we call traditional reward—predefined and necessarily Markovian—"MDP reward", contrasting with human reward.

[5]The one exception, a mountain car trainer, gave an equal amount of positive and negative reward. The Tetris agents of eight trainers could not clear even 10 lines a game, in many cases averaging less than a line cleared per game. Yet these trainers still gave more positive reward than negative reward, despite dreadful task performance.

[6]Though giving negative values to all human reward would communicate that the task is goal-based, this mapping is not an option because we seek algorithms that are agnostic to whether the task is goal-based.

[7]This algorithm estimates return for each possible immediate action by taking the highest-return path on that action's branch, where a path's return is calculated as the sum of discounted reward along the path and the discounted, learned return at the leaf state of the path. Action selection is similar to ϵ-greedy: there is a probability ϵ at each step that the agent will choose a uniformly random action, and otherwise the action is that with the highest estimated return. Lastly, the depth of the agent's exhaustive tree search is chosen from a Uniform(0,3) distribution at each step to provide a wider range of experiences. The agent updates its value function only on experienced transitions. The Sarsa(λ) parameters are below, following Sutton and Barto's notation [14]. The action-value function Q is represented by a linear model over Gaussian RBF features. For each action, 1600 RBF means are located on a 40×40 evenly spaced grid over the state space, where the outermost means in each dimension lie on the extremes of the dimension. Additionally, an activation feature of 0.1 is added for each action, creating a total of 4803 state-action features. When an action is input to Q, the features for all other actions are zero. The width σ^2 of the Gaussian RBFs is 0.08, following Sutton and Barto's definition of an RBF's "width" and where the unit is the distance in normalized state space between adjacent Gaussian means. All weights of Q are optimistically initialized to 0. The Sarsa(λ) algorithm uses ϵ-greedy action selection, starting with $\epsilon = 0.1$ and annealing ϵ after each episode by a factor of 0.998. Eligibility traces were created as replacing traces with $\lambda = 0.84$. The step size $\alpha = 0.01$.

[8]Each $\hat{R}_H$ trained on the first 20 episodes of its corresponding training log. To account for a small step size (0.001), each $\hat{R}_H$ was trained from 100 epochs on the trainer log. Credit assignment was performed by the "aggregate reward" method, updating only when reward was received as in the "reward-only" condition described in Section 3.4.3 of Knox's dissertation [6].

[9]In general, the choice of RL algorithm will impact performance, so one might ask whether the algorithm used here is actually learning an MDP-optimal policy for its corresponding human reward model and discount factor. At $\gamma = 0$ and $\gamma = 1$, the answer appears to be "yes." At $\gamma = 0$, the agent optimizes return at tree search depths greater than 0. When the search depth is zero, it uses the learned value for Q(s,a), which is roughly equivalent to $\hat{R}_H(s, a)$ after many learning samples at or near (s,a). At $\gamma = 1$, if the RL algorithm learns an infinitely repeatable sequence of actions with positive net reward, then the disastrous policy that loops on that sequence is necessarily within the set of MDP-optimal policies (with respect to predictions of human reward). As mentioned above, we visually checked the behavior of five models' corresponding algorithms while they exhibited the worst possible performance, and each agent looped until the episode limit was reached. During looping, the maximum Q values at all observed states were positive. Therefore, the results for $\gamma = 0$ and $\gamma = 1$ can be considered correct — independent of the RL algorithm used — with confidence. However, for $0 < \gamma < 1$, another RL algorithm might learn a policy with a higher mean $R_{\hat{H}}(s_o)$ than the mean return in these results.

[10]Each RBF has a width $\sigma^2 = 0.05$, where 1 is the distance to the nearest adjacent RBF center, and the linear model has an additional bias feature of constant value 0.1. $\hat{R}_H$ is updated with new feedback by incremental gradient descent with a step size of 0.2. In accordance with the most recent version of TAMER [6], we used aggregate reward for credit assignment with a probability distribution over feedback delay of Uniform(-0.4 seconds, -0.15 seconds) (with negative values because the algorithm looks backwards in time from the feedback signal to potentially targeted events), and updates occurred at every step regardless of whether reward was provided.

[11]Variation in subject numbers comes from a few user errors (usually not typing in their experimental condition correctly), errors in logging, and the removal of 3 subjects for insufficient feedback (the 1 removed subject who gave any feedback had a feedback-instances-to-time-steps ratio of 0.01.; subjects who were retained had ratios above 0.1).